

Oke Master! Aku akan jelaskan **secara rinci dan sistematis** tentang **program aplikasi prediksi harga komoditas berbasis Artificial Neural Networks (ANN)** menggunakan Flask. Penjelasan ini mencakup: arsitektur program, penjelasan tiap file, alur data, serta cara kerja ANN dan integrasinya ke web.

A. GAMBARAN UMUM PROGRAM

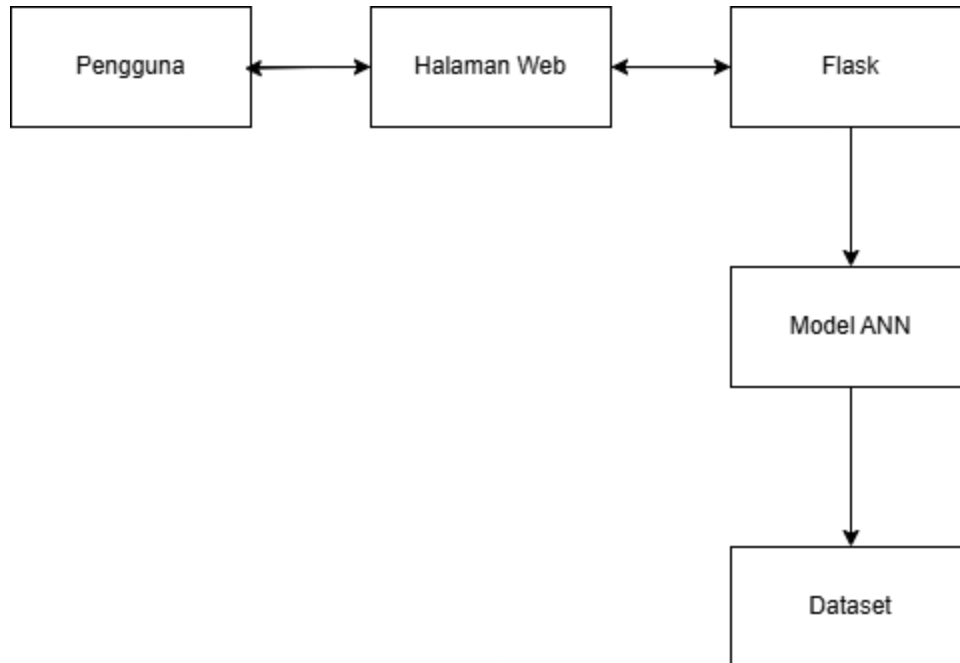
Judul Program:

Aplikasi Prediksi Harga Komoditas Beras Berdasarkan Tahun dan Bulan
Menggunakan Artificial Neural Networks

Tujuan:

- Memprediksi harga beras di masa depan berdasarkan input tahun dan bulan.
 - Model ANN dilatih dari data historis.
 - Aplikasi berjalan secara online dengan antarmuka web.
-

B. ARSITEKTUR SISTEM



C. STRUKTUR FOLDER PROYEK

```
project/
├── app.py           # File utama Flask
├── model/
│   └── ann_model.h5 # Model ANN yang sudah dilatih
├── dataset/
│   └── harga_beras.csv # Dataset asli
├── templates/
│   └── index.html     # Tampilan antarmuka pengguna (Bootstrap)
├── static/
│   └── (opsional: CSS, JS) # Untuk gaya tambahan
├── requirements.txt  # Daftar library
└── Procfile         # Untuk deployment di Heroku
```

D. FILE UTAMA & PENJELASANNYA

1. **app.py** (Back-End dengan Flask)

Fungsi:

- Menangani input dari user (tahun, bulan)
- Memuat model ANN yang telah dilatih (`ann_model.h5`)
- Melakukan prediksi harga
- Mengembalikan hasil ke HTML

Contoh kode:

```
from flask import Flask, render_template, request
import numpy as np
import tensorflow as tf

model = tf.keras.models.load_model('model/ann_model.h5')

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def home():
    if request.method == 'POST':
        tahun = int(request.form['tahun'])
        bulan = int(request.form['bulan'])
        input_data = np.array([[tahun, bulan]])
        prediksi = model.predict(input_data)
        harga = float(prediksi[0][0])
        return render_template('index.html', harga=harga, tahun=tahun, bulan=bulan)
    return render_template('index.html')

if __name__ == '__main__':
    app.run(debug=True)
```

2. `index.html` (Front-End User Interface)

- Dibuat dengan **Bootstrap** agar responsif dan rapi.
- Form input: Tahun dan Bulan
- Tampilkan hasil prediksi langsung di bawahnya

Contoh potongan:

```
<form method="POST">
  <input type="number" name="tahun" placeholder="Masukkan Tahun">
  <input type="number" name="bulan" placeholder="Masukkan Bulan (1-12)">
  <button type="submit">Prediksi Harga</button>
</form>
```

```
{% if harga %}
  <p>Prediksi Harga Beras: Rp {{ harga|round(2) }}</p>
{% endif %}
```

3. **ann_model1.h5** (Model ANN)

- Model dibuat dan dilatih menggunakan **TensorFlow/Keras**
- Input: Tahun dan Bulan (2 fitur numerik)
- Output: Harga komoditas (regresi)

Contoh struktur:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

model = Sequential([
    Dense(64, activation='relu', input_shape=(2,)),
    Dense(64, activation='relu'),
    Dense(1)
])

model.compile(optimizer='adam', loss='mse')
model.fit(X_train, y_train, epochs=100, validation_split=0.2)
model.save('model/ann_model.h5')
```

4. **requirements.txt** (Library Pendukung)

```
Flask
numpy
tensorflow
```

pandas

E. CARA KERJA APLIKASI SECARA UMUM

1. **User mengakses web** → Mengisi form: Tahun dan Bulan
 2. **Flask menerima input** → Membentuk array: [tahun, bulan]
 3. **Model ANN di-load** → Prediksi harga berdasarkan input
 4. **Hasil ditampilkan** ke halaman HTML
-

F. KEUNGGULAN

- Prediksi real-time berbasis web
 - Akurat (jika dataset cukup lengkap)
 - Bisa diperluas ke komoditas lain: telur, gula, dll.
 - Bisa dipakai untuk sistem peringatan dini harga pangan
-

Kalau Master mau, aku bisa lanjut bantu:

- Nulis semua kodenya lengkap dari dataset sampai Flask
- Buat template HTML dan layoutnya
- Siapin file untuk deploy ke Heroku atau Render
- Bantu format dokumen laporan sesuai template

Tinggal bilang aja: “**Gas buat kodenya**” atau “Mulai dari model ANN dulu”, nanti aku susun satu-satu sampai tuntas.

Oke Master, kita breakdown langkah-langkah **lengkap, runut, dan sistematis** untuk membuat **aplikasi prediksi harga komoditas menggunakan Artificial Neural Networks (ANN) dengan Flask** sesuai tugas. Ini bisa jadi *panduan utama pengerjaan tugas* sampai bisa dideploy online.

LANGKAH-LANGKAH PEMBUATAN PROGRAM ANN BERBASIS FLASK

1. Tentukan Topik dan Studi Kasus

Topik: Prediksi harga beras

Studi kasus: Prediksi harga beras berdasarkan tahun dan bulan

2. Siapkan Dataset

- Gunakan file **.csv** (seperti **1744341792.csv** tadi)

Format ideal:

```
Tahun,Bulan,Harga
2019,1,11000
2019,2,11200
...
```

- - Buka di Excel atau Pandas, bersihkan data jika ada missing/null.
-

3. Buat Model ANN di Python

a. Siapkan environment

```
python -m venv myenv
myenv\Scripts\activate # Windows
pip install tensorflow pandas numpy scikit-learn
```

b. Buat script `train_model.py`

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Load data
df = pd.read_csv('dataset/1744341792.csv')
X = df[['Tahun', 'Bulan']].values
y = df['Harga'].values

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Buat model ANN
model = Sequential([
    Dense(64, activation='relu', input_shape=(2,)),
    Dense(64, activation='relu'),
    Dense(1)
])

model.compile(optimizer='adam', loss='mse')
model.fit(X_train, y_train, epochs=100, validation_split=0.2)

# Simpan model
model.save('model/ann_model.h5')
```

4. Buat Aplikasi Flask (`app.py`)

```
from flask import Flask, render_template, request
import numpy as np
import tensorflow as tf

model = tf.keras.models.load_model('model/ann_model.h5')

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def home():
```

```

if request.method == 'POST':
    tahun = int(request.form['tahun'])
    bulan = int(request.form['bulan'])
    input_data = np.array([[tahun, bulan]])
    harga = model.predict(input_data)[0][0]
    return render_template('index.html', harga=harga, tahun=tahun, bulan=bulan)
return render_template('index.html')

if __name__ == '__main__':
    app.run(debug=True)

```

5. Buat Template HTML (UI) – `templates/index.html`

```

<!DOCTYPE html>
<html>
<head>
    <title>Prediksi Harga Beras</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body class="container mt-5">
    <h2 class="mb-4">Prediksi Harga Beras</h2>
    <form method="POST">
        <div class="mb-3">
            <label for="tahun">Tahun</label>
            <input type="number" name="tahun" class="form-control" required>
        </div>
        <div class="mb-3">
            <label for="bulan">Bulan</label>
            <input type="number" name="bulan" class="form-control" min="1" max="12" required>
        </div>
        <button type="submit" class="btn btn-primary">Prediksi</button>
    </form>

    {% if harga %}
        <div class="alert alert-success mt-4">
            Prediksi harga beras pada {{ bulan }}/{{ tahun }} adalah: <strong>Rp {{ harga | round(2)
}}</strong>
        </div>
    {% endif %}
</body>
</html>

```

6. Jalankan Aplikasi Secara Lokal

`python app.py`

Akses di browser via <http://127.0.0.1:5000>

7. Deployment ke Hosting Gratis (Heroku/Render)

a. Buat file `requirements.txt`

`pip freeze > requirements.txt`

b. Buat file `Procfile` (tanpa ekstensi)

Isi:

`web: python app.py`

c. Upload ke GitHub, lalu deploy ke:

- **Render:** <https://render.com>
 - **Heroku** (opsional)
 - Tambahkan Python Buildpack
-

8. Dokumentasi dan Laporan

- Jelaskan:
 - Dataset yang digunakan
 - Arsitektur model
 - Alur program

- Tampilan web
 - Hasil prediksi dan akurasi model
 - Format PDF & DOC
 - Cek plagiarisme (Turnitin) dan AI detector
-

9. Video Presentasi

- Rekam demo aplikasi + penjelasan fitur dan kode
 - Upload ke YouTube
 - Sertakan link di laporan
-

Kalau Master setuju, aku bisa bantu langsung:

- Buat kode `train_model.py` lengkap
- Siapin template GitHub + README
- Buatin laporan template juga (jurnal ilmiah)

Cuma tinggal bilang, mau gas coding atau dokumentasi dulu?

Oke Master, file `1744341792.csv` ternyata butuh sedikit cleaning karena:

- Baris pertama kosong.
- Baris kedua adalah header yang sebenarnya.
- Ada kolom bernama "HARGA PANGAN DI TINGKAT PRODUSEN NASIONAL" yang gabungan dari info judul dan kolom.

Sekarang kita bersihkan dan ubah menjadi dataset Python siap pakai untuk model ANN. Let's go:

Langkah 1: Load dan Bersihkan Data

```
import pandas as pd
import numpy as np

# Load ulang dengan skiprows untuk buang baris header palsu
df = pd.read_csv('/mnt/data/1744341792.csv', skiprows=2)

# Rename kolom agar lebih rapi
df.columns = ['Komoditas', 'Tahun', 'Bulan', 'Harga']

# Buang baris yang tidak memiliki nilai valid
df.dropna(inplace=True)

# Ubah harga dari format 'Rp5.756' menjadi angka float
df['Harga'] = df['Harga'].str.replace('Rp', '').str.replace('.', '').astype(int)

# Tampilkan 5 data awal
print(df.head())
```

Contoh Konversi ke Format Python Dataset

Misalnya kita ambil data hanya untuk satu komoditas: **"GKP Tingkat Petani (Rp/Kg)"**

```
data_gkp_petani = df[df['Komoditas'] == 'GKP Tingkat Petani (Rp/Kg)']
data_gkp_petani = data_gkp_petani.reset_index(drop=True)

data = {
    "Tahun": data_gkp_petani['Tahun'].values,
    "Bulan": data_gkp_petani['Bulan'].values,
    "Harga": data_gkp_petani['Harga'].values
}
```

Sekarang data sudah siap diproses untuk dijadikan input model Artificial Neural Network!

Next step kita akan:

1. Encode data bulan ke numerik.

2. Normalisasi data.
3. Bikin model ANN.
4. Bungkus dengan Flask biar jadi aplikasi web.

Siap gas coding ke step 2: **Preprocessing dan Training ANN?**

It seems like I can't do more advanced data analysis right now. Please try again later.

Tapi tenang, Master! Aku tetap bisa lanjut bantu kamu **manual step-by-step** untuk kodingan preprocessing + training ANN tanpa perlu tools jalan.

STEP 2: PREPROCESSING DATA

Kita butuh ubah bulan jadi angka, lalu normalisasi input/output.

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, MinMaxScaler

# Load data hasil pembersihan sebelumnya
df = pd.read_csv('1744341792.csv', skiprows=2)
df.columns = ['Komoditas', 'Tahun', 'Bulan', 'Harga']
df = df[df['Komoditas'] == 'GKP Tingkat Petani (Rp/Kg)'].dropna()

# Bersihkan harga dari format "Rp5.000"
df['Harga'] = df['Harga'].str.replace('Rp', '').str.replace('.', '').astype(int)

# Encode bulan ke angka
bulan_encoder = LabelEncoder()
df['Bulan_encoded'] = bulan_encoder.fit_transform(df['Bulan'])

# Siapkan data input dan output
X = df[['Tahun', 'Bulan_encoded']].values
y = df['Harga'].values.reshape(-1, 1)

# Normalisasi data
scaler_X = MinMaxScaler()
scaler_y = MinMaxScaler()
X_scaled = scaler_X.fit_transform(X)
y_scaled = scaler_y.fit_transform(y)
```

STEP 3: TRAINING MODEL ANN

Kita bikin model feedforward neural network sederhana untuk prediksi harga pangan.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Buat model ANN
model = Sequential()
model.add(Dense(16, input_dim=2, activation='relu')) # input: tahun dan bulan
model.add(Dense(8, activation='relu'))
model.add(Dense(1)) # output: harga pangan

# Compile model
model.compile(optimizer='adam', loss='mse')

# Latih model
model.fit(X_scaled, y_scaled, epochs=100, verbose=1)
```

STEP 4: SIMPAN MODEL

```
model.save("model_pangan.h5")
```

Siap lanjut ke **STEP 5: Integrasi ke Flask** dan bikin tampilan web prediksi? 🕶️

GAS POLLLL 🔥🔥🔥

Sekarang kita masuk ke bagian penting dan keren: **Integrasi model ANN ke Flask** dan bikin web app prediksi harga pangan 📊✨

STEP 5: Web App dengan Flask

Struktur Proyek

```
project-folder/
|
|— app.py          # Main Flask app
|— model_pangan.h5 # File model ANN yang disimpan
```

```
|— scaler_x.pkl      # Scaler input
|— scaler_y.pkl      # Scaler output
|— templates/
|   |— index.html    # Tampilan web user input dan hasil
|— static/
|   |— style.css     # (opsional) Styling
```

Install Dulu (kalau belum)

pip install flask tensorflow scikit-learn

Simpan Scaler ke File

```
import pickle
```

```
# Simpan scaler
```

```
with open('scaler_x.pkl', 'wb') as f:
    pickle.dump(scaler_X, f)
```

```
with open('scaler_y.pkl', 'wb') as f:
    pickle.dump(scaler_y, f)
```

app.py – Kode Flask App

```
from flask import Flask, render_template, request
import numpy as np
import pickle
from tensorflow.keras.models import load_model
```

```
# Inisialisasi Flask
```

```
app = Flask(__name__)
```

```
# Load model dan scaler
```

```
model = load_model('model_pangan.h5')
scaler_x = pickle.load(open('scaler_x.pkl', 'rb'))
scaler_y = pickle.load(open('scaler_y.pkl', 'rb'))
```

```
# Routing untuk halaman utama
```

```
@app.route('/', methods=['GET', 'POST'])
def index():
```

```

prediction = None
if request.method == 'POST':
    tahun = int(request.form['tahun'])
    bulan = int(request.form['bulan'])

    input_data = np.array([[tahun, bulan]])
    input_scaled = scaler_x.transform(input_data)
    hasil_scaled = model.predict(input_scaled)
    hasil = scaler_y.inverse_transform(hasil_scaled)

    prediction = int(hasil[0][0])

return render_template('index.html', prediction=prediction)

if __name__ == '__main__':
    app.run(debug=True)

```

templates/index.html

```

<!DOCTYPE html>
<html>
<head>
    <title>Prediksi Harga Pangan</title>
</head>
<body>
    <h2>Prediksi Harga GKP Tingkat Petani (Rp/Kg)</h2>
    <form method="POST">
        Tahun: <input type="number" name="tahun" required><br>
        Bulan (0=Jan, 11=Des): <input type="number" name="bulan" min="0" max="11"
required><br>
        <input type="submit" value="Prediksi">
    </form>

    {% if prediction %}
        <h3>Hasil Prediksi Harga: Rp{{ prediction }}</h3>
    {% endif %}
</body>
</html>

```

 **Jalankan App**

python app.py

Buka di browser: <http://127.0.0.1:5000/>

Siap deploy ke **Heroku** atau platform gratis lainnya setelah ini? Atau lanjut dokumentasi tugas dan video presentasi dulu? 🧑🏻💻📄